

Hopf-algebraic structures of families of trees

Robert Grossman*

University of California, Berkeley

Richard G. Larson†

University of Illinois at Chicago

February, 1987

This is a draft of a paper which subsequently appeared in the Journal of Algebra, Volume 26, 1989, pp. 184–210.

1 Introduction

In this paper we describe Hopf algebras which are associated with certain families of trees. These Hopf algebras originally arose in a natural fashion: one of the authors [5] was investigating data structures based on trees, which could be used to efficiently compute certain differential operators. Given data structures such as trees which can be multiplied, and which act as higher-order derivations on an algebra, one expects to find a Hopf algebra of some sort. We were pleased to find that not only was there a Hopf algebra associated with these data structures, but that it could be used to give new proofs of enumerations of such objects as rooted trees and ordered rooted trees. Previous work applying Hopf algebras to combinatorial objects (such as [10], [13] or [14]) has concerned itself with algebraic structures on polynomial algebras and on partially ordered sets, rather than on trees themselves.

*The first author is a National Science Foundation Postdoctoral Research Fellow.

†This paper was written while the second author was on sabbatical leave at the University of California, Berkeley.

We hope that these constructions will also provide insight for the algebra of data structures.

The Hopf algebras which we construct are all cocommutative graded connected Hopf algebras. This allows us to apply the Milnor-Moore Theorem (Theorem 3.4) and the Poincaré-Birkhoff-Witt Theorem (Theorem 3.3) to get precise information on the structure of these Hopf algebras. We illustrate our construction, and its application, by sketching a proof of Cayley's enumeration [3] of finite rooted trees.

We now describe how to construct the Hopf algebra $k\{\mathcal{T}\}$ which has as basis all finite rooted trees. The grading on $k\{\mathcal{T}\}$ is given as follows: if the tree t has $n + 1$ nodes, then $t \in k\{\mathcal{T}\}_n$. If t_1 and t_2 are trees, the product $t_1 \cdot t_2$ is the sum of the trees formed by attaching the children of the root of t_1 to the nodes of t_2 in all possible ways. If t is a tree, the coproduct $\Delta(t)$ is the sum of all terms $t_1 \otimes t_2$, where the children of the root of t_1 and the children of the root of t_2 range over all possible partitions of the children of the root of t . This definition of the coproduct is very similar to the definition of the coproduct in the *placement coalgebra* described in [10]. Both here and in [10], the coproduct of a structure is the sum of all terms which are the tensor product of the two pieces resulting from decomposing in all possible ways the structure into two disjoint substructures.

We show in Sections 2 and 3 that this product and coproduct gives a cocommutative graded connected Hopf algebra on the vector space which has as basis the isomorphism classes of finite rooted trees. Specifically, in Section 2 we give axioms which a family of trees must satisfy to support a Hopf algebra structure. In Section 3 we define the Hopf algebra associated with such a family. In our example, it is straightforward to show (Theorem 4.1) that $P(k\{\mathcal{T}\})$, the space of primitive elements in $k\{\mathcal{T}\}$, has as basis the set of trees whose root has exactly one child. Therefore we have a one-one correspondence between a basis for $k\{\mathcal{T}\}_{n-1}$ and a basis for $P(k\{\mathcal{T}\})_n$: any tree with n nodes corresponds to the tree with $n + 1$ nodes formed by creating a new root and linking the root of the original tree to it as a child. If we let t_n be the number of trees with n nodes, $a_n = \dim k\{\mathcal{T}\}_n$, and $p_n = \dim P(k\{\mathcal{T}\})_n$, we have that $t_{n+1} = a_n$ and $p_n = a_{n-1}$. The Milnor-Moore Theorem and the Poincaré-Birkhoff-Witt Theorem imply that

$$a_n = \sum_{m_1+2m_2+\dots+rm_r=n} \binom{m_1+p_1-1}{m_1} \dots \binom{m_r+p_r-1}{m_r}$$

$$= \sum_{m_1+2m_2+\dots+rm_r=n} \binom{m_1+a_0-1}{m_1} \dots \binom{m_r+a_{r-1}-1}{m_r},$$

which implies that

$$t_{n+1} = \sum_{m_1+2m_2+\dots+rm_r=n} \binom{m_1+t_1-1}{m_1} \dots \binom{m_r+t_r-1}{m_r},$$

which implies, using

$$\binom{-r}{k} = (-1)^k \binom{r+k-1}{k},$$

Cayley's result that if $T(z) = \sum_{n=1}^{\infty} t_n z^n$ then

$$T(z) = z \prod_{n=1}^{\infty} (1 - z^n)^{-t_n}.$$

In Section 4 we give the proof we have just sketched, in full detail, for labeled (or colored) trees. In Section 5 we discuss the algebraic structure of the Hopf algebras constructed for the family of labeled ordered trees (that is, for labeled or colored trees in which the children of each root are linearly ordered). In Section 6 we discuss the algebraic structure of the Hopf algebras constructed for the family of heap-ordered trees, and for related families. Heap-ordered trees (see [16] for details) are used as data structures in computer science and symbolic algebra. In this section, we also give a brief discussion of the category of families of trees.

Throughout this paper, the field k will have characteristic 0.

2 The structure of families of rooted trees

In this section we describe a structure we impose on families of rooted trees. By a tree we mean a nonempty finite rooted tree, and by a forest we mean a finite family of finite rooted trees, possibly empty. The mathematical structure \mathcal{X} consists of a family $\text{Tree}(\mathcal{X})$ of trees, and a family $\text{Forest}(\mathcal{X})$ of forests, both possibly with additional structure, together with four operations. In most of the examples we consider, the additional structure consists of orderings or labels. As we proceed, we impose axioms requiring that the operations preserve this additional structure. The operations are:

- an operation DelRoot which maps $\text{Tree}(\mathcal{X})$ to $\text{Forest}(\mathcal{X})$. This sends a tree $t \in \text{Tree}(\mathcal{X})$ to the forest obtained when the root of t is deleted. Note that the exact definition of DelRoot depends upon the additional structure we impose on $\text{Tree}(\mathcal{X})$ and $\text{Forest}(\mathcal{X})$.
- an operation Nodes which maps $\text{Tree}(\mathcal{X})$ to **Sets**. This sends a tree $t \in \text{Tree}(\mathcal{X})$ to the set consisting of the nodes of the underlying tree of t .
- a restriction operation which maps subforests of $\text{Forest}(\mathcal{X})$ to $\text{Forest}(\mathcal{X})$. If $V \in \text{Forest}(\mathcal{X})$, and if $U \subseteq V$ is a subforest, then we must specify a way to impose the additional structure of \mathcal{X} on the forest U in a manner reflecting the structure of V . We denote the resulting element of $\text{Forest}(\mathcal{X})$ by $V|U$.
- an attachment operation which is a map from $\text{Forest}(\mathcal{X}) \times \text{Tree}(\mathcal{X})$ to $\text{Tree}(\mathcal{X})$. We denote this operation by \leftarrow . If $t \in \text{Tree}(\mathcal{X})$, $U \in \text{Forest}(\mathcal{X})$, and $\mathbf{d} : U \rightarrow \text{Nodes}(t)$, then $t \leftarrow_{\mathbf{d}} U \in \text{Tree}(\mathcal{X})$. Intuitively, what \leftarrow does is the following: it forms a new tree by linking the root of each tree u in the forest U to the node $\mathbf{d}(u)$ of t , in a manner which preserves the additional structure of \mathcal{X} .

We give some examples of specific families of trees we consider in this paper. For each of these examples, it is easy to check that the nine axioms below are satisfied.

Example 2.1 *The simplest example is \mathcal{T} , the family of trees without any additional structure. The set $\text{Tree}(\mathcal{T})$ is the set of finite rooted trees. The set $\text{Forest}(\mathcal{T})$ is the set of finite forests of finite rooted trees. The map DelRoot sends each tree into the forest formed by deleting its root. If $V \in \text{Forest}(\mathcal{T})$ and $U \subset V$ is a subforest, then $V|U$ is U . If t is a finite rooted tree, U is a finite forest of finite rooted trees, and $\mathbf{d} : U \rightarrow \text{Nodes}(t)$ is a function, then $t \leftarrow_{\mathbf{d}} U$ is the tree formed by linking the root of each tree in U to the node $\mathbf{d}(u)$ of t .*

Most of the axioms we give for the family \mathcal{X} will consist of assertions that facts we explicitly prove for \mathcal{T} hold for \mathcal{X} .

Example 2.2 *The family \mathcal{OT} of ordered trees. The set $\text{Tree}(\mathcal{OT})$ is the set of finite rooted trees, together with total orderings on the sets of children of*

each node. The set $\text{Forest}(\mathcal{OT})$ is the set of finite forests of finite rooted trees, together with total orderings on the sets of children of each node, and a total ordering on each forest. The map DelRoot sends each tree into the forest formed by deleting its root. The ordering on the forest is the one arising from the ordering on the children of the root of the original tree. If $V \in \text{Forest}(\mathcal{OT})$ and $U \subset V$ is a subforest, then $V|U$ is U , with the same orderings on the sets of children of each node as in V , and with the ordering on the trees in U induced as a subset of the trees of V . If t is an ordered finite rooted tree, U is an ordered finite forest of ordered finite rooted trees, and $\mathbf{d} : U \rightarrow \text{Nodes}(t)$ is a function, then $t \leftarrow_{\mathbf{d}} U$ is the tree formed by linking the root of each tree in U to the node $\mathbf{d}(u)$ of t . The ordering on the children of a node α of $t \leftarrow_{\mathbf{d}} U$ is given as follows: the children newly linked to α follow all of the original children of α in the new order, and keep the order among themselves which was induced by the ordering of the forest U ; the original children of α keep their original order.

Note that in the above example, nodes which do not have a common parent are not related by any order. Also note that we have orderings on subsets of sets of nodes of trees, and also have orderings on forests of trees.

Example 2.3 The family \mathcal{HOT} of heap-ordered trees. The set $\text{Tree}(\mathcal{HOT})$ is the set of finite rooted trees, together with a total ordering on the set of all nodes of each tree, such that each node precedes all of its children in the ordering. The set $\text{Forest}(\mathcal{HOT})$ is the set of finite forests of finite rooted trees, together with a total ordering on the set of all the nodes of each forest, such that each node precedes all of its children in the ordering. The map DelRoot sends each tree into the forest formed by deleting its root. The ordering on the nodes of the forest is the one arising from the ordering on the nodes of the original tree. If $V \in \text{Forest}(\mathcal{HOT})$ and $U \subset V$ is a subforest, then $V|U$ is U , with the ordering on the nodes of U being induced as a subset of the nodes of V . If $t \in \text{Tree}(\mathcal{HOT})$, $U \in \text{Forest}(\mathcal{HOT})$, and $\mathbf{d} : U \rightarrow \text{Nodes}(t)$ is a function, then $t \leftarrow_{\mathbf{d}} U$ is the tree formed by linking the root of each tree in U to the node $\mathbf{d}(u)$ of t . The new ordering on the nodes of $t \leftarrow_{\mathbf{d}} U$ is given as follows: the nodes in the forest U follow all of the nodes of the tree t in the new order, and keep the order among themselves which was induced by the ordering of the nodes of the trees of the forest U ; the original nodes in t keep their original order.

Example 2.4 The family $\mathcal{LT}(E_1, \dots, E_M)$ of labeled trees, where $\{E_1, \dots, E_M\}$ is a set of formal symbols. The set $\text{Tree}(\mathcal{LT}(E_1, \dots, E_M))$ is the set of finite rooted trees, together with an assignment of an element of $\{E_1, \dots, E_M\}$ to each node, other than the root, of each tree. The set $\text{Forest}(\mathcal{LT}(E_1, \dots, E_M))$ is the set of finite forests of finite rooted trees, together with an assignment of an element of $\{E_1, \dots, E_M\}$ to each node in the forest. The map DelRoot sends each tree into the forest formed by deleting its root. The assignment of symbols to the nodes is not changed. If $V \in \text{Forest}(\mathcal{LT}(E_1, \dots, E_M))$ and $U \subset V$ is a subforest, then $V|U$ is U with the same assignment of formal symbols. If $t \in \text{Tree}(\mathcal{LT})$, $U \in \text{Forest}(\mathcal{LT})$, and $\mathbf{d} : U \rightarrow \text{Nodes}(t)$ is a function, then $t \leftarrow_{\mathbf{d}} U$ is the tree formed by linking the root of each tree in U to the node $\mathbf{d}(u)$ of t . The assignment of symbols to the nodes is not changed.

The families \mathcal{LOT} of labeled ordered trees, and \mathcal{LHOT} of labeled heap-ordered trees, are defined analogously.

Labeled trees are called *colored* trees by some authors. Note that for $\mathcal{X} = \mathcal{LT}$, \mathcal{LOT} , or \mathcal{LHOT} , the trees in $\text{Tree}(\mathcal{X})$ do not have their roots labeled, but the trees in forests in $\text{Forest}(\mathcal{X})$ have all of their nodes, including their roots, labeled. The above examples all occur in applications to analysis and to data structures used in symbolic algebra.

The first axiom avoids degenerate families \mathcal{X} .

Axiom 1 Every finite rooted tree occurs as the underlying tree of some element of $\text{Tree}(\mathcal{X})$. Every finite forest of finite rooted trees appears as the underlying forest of some element of $\text{Forest}(\mathcal{X})$.

The next three axioms describe how the operation $|$ relates to the structure of elements of $\text{Forest}(\mathcal{X})$.

Axiom 2 Let $V \in \text{Forest}(\mathcal{X})$. Then $V|V = V$.

Axiom 3 Let $V \in \text{Forest}(\mathcal{X})$, and let $U \subset V$ be a subforest. Then the underlying forest of $V|U$ is U .

Axiom 4 Let $W \in \text{Forest}(\mathcal{X})$, and let $U \subset V \subset W$ be subforests. Then

$$W|U = (W|V)|U.$$

The next axiom describes how the linking operation \leftarrow interacts with the structure of the elements of $\text{Tree}(\mathcal{X})$ and $\text{Forest}(\mathcal{X})$.

Axiom 5 *Let $t \in \text{Tree}(\mathcal{X})$, $U \in \text{Forest}(\mathcal{X})$, and $\mathbf{d} : U \rightarrow \text{Nodes}(t)$. Then $t \leftarrow_{\mathbf{d}} U \in \text{Tree}(\mathcal{X})$. The underlying tree of $t \leftarrow_{\mathbf{d}} U$ is the tree formed by linking the root of each tree $u \in U$ to the node $\mathbf{d}(u)$ of t .*

Axiom 6 *There exists a unique $e \in \text{Tree}(\mathcal{X})$ such that the underlying tree of e has exactly one node.*

There exists a unique $\emptyset \in \text{Forest}(\mathcal{X})$ such that the underlying forest of \emptyset is the empty set.

For any $U \in \text{Forest}(\mathcal{X})$ there is a unique function

$$\mathbf{1} : U \rightarrow \text{Nodes}(e).$$

For any $t \in \text{Tree}(\mathcal{X})$ there is a unique function

$$\mathbf{0} : \emptyset \rightarrow \text{Nodes}(t).$$

Axiom 7 *If $t \in \text{Tree}(\mathcal{X})$, then*

$$t = e \leftarrow_{\mathbf{1}} \text{DelRoot}(t),$$

and

$$t = t \leftarrow_{\mathbf{0}} \emptyset.$$

If $U \in \text{Forest}(\mathcal{X})$, then

$$U = \text{DelRoot}(e \leftarrow_{\mathbf{1}} U).$$

The previous axiom implies that there is redundancy in the way we have defined our structure \mathcal{X} : $\text{Tree}(\mathcal{X})$ and $\text{Forest}(\mathcal{X})$ are essentially isomorphic via the maps $t \mapsto \text{DelRoot}(t)$ and $U \mapsto (e \leftarrow_{\mathbf{1}} U)$. However, we have chosen to retain both $\text{Tree}(\mathcal{X})$ and $\text{Forest}(\mathcal{X})$ for purposes of clarity and intuitiveness.

If we take three finite rooted trees t_1 , t_2 , and t_3 , there are two ways to form a tree: with the children of the root of t_1 linked to the nodes of t_2 , and with the children of the root of t_2 linked to the nodes of t_3 . We show that these two ways are the same. (This is essentially an associativity condition.)

Suppose that

$$\mathbf{d} : \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_2),$$

and

$$\mathbf{e} : \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)) \rightarrow \text{Nodes}(t_3)$$

are linking functions. Then we can form the tree

$$t_3 \leftarrow_{\mathbf{e}} \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)).$$

From these data we can construct linking functions

$$\mathbf{f} : \text{DelRoot}(t_2) \rightarrow \text{Nodes}(t_3)$$

and

$$\mathbf{g} : \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2))$$

as follows. Every tree in the forest $\text{DelRoot}(t_2)$ corresponds to a tree in the forest $\text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1))$. Let

$$\iota : \text{DelRoot}(t_2) \rightarrow \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)) \quad (2.5)$$

be the injection which gives this correspondence. The function \mathbf{f} is given by

$$\mathbf{f} = \mathbf{e} \circ \iota.$$

The function \mathbf{g} is given by

$$\mathbf{g}(s) = \begin{cases} \mathbf{d}(s) & \text{if } \mathbf{d}(s) \neq \text{root}(t_2) \\ \mathbf{e}(s) & \text{if } \mathbf{d}(s) = \text{root}(t_2). \end{cases}$$

Note that we identify $\text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1))$ with the disjoint union

$$\iota(\text{DelRoot}(t_2)) \cup \{ u \in \text{DelRoot}(t_1) \mid \mathbf{d}(u) = \text{root}(t_2) \},$$

and identify $\text{Nodes}(t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2))$ with the disjoint union of $\text{Nodes}(t_3)$ and $\text{Nodes}(t_2) \setminus \{\text{root}(t_2)\}$. ($X \setminus Y$ denotes the set-theoretic difference of X and Y .) We denote the map sending the pair (\mathbf{d}, \mathbf{e}) to the pair (\mathbf{f}, \mathbf{g}) as follows:

$$\mathbf{R}(\mathbf{d}, \mathbf{e}) = (\mathbf{f}, \mathbf{g}).$$

Suppose now that

$$\mathbf{f} : \text{DelRoot}(t_2) \rightarrow \text{Nodes}(t_3),$$

and

$$\mathbf{g} : \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2))$$

are linking functions. Then we can form the tree

$$(t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2)) \leftarrow_{\mathbf{g}} \text{DelRoot}(t_1).$$

From these data we can construct linking functions

$$\mathbf{d} : \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_2)$$

and

$$\mathbf{e} : \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)) \rightarrow \text{Nodes}(t_3)$$

as follows. The function \mathbf{d} is given by

$$\mathbf{d}(s) = \begin{cases} \mathbf{g}(s) & \text{if } \mathbf{g}(s) \in \text{Nodes}(t_2) \setminus \{\text{root}(t_2)\} \\ \text{root}(t_2) & \text{otherwise.} \end{cases}$$

The function \mathbf{e} is given by

$$\mathbf{e}(s) = \begin{cases} \mathbf{f}(s') & \text{if } s = \iota(s') \text{ with } s' \in \text{DelRoot}(t_2) \\ \mathbf{g}(s) & \text{if } s \in \text{DelRoot}(t_1) \text{ and } \mathbf{g}(s) \in \text{Nodes}(t_3). \end{cases}$$

Note that we identify $\text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1))$ with the disjoint union

$$\iota(\text{DelRoot}(t_2)) \cup \{u \in \text{DelRoot}(t_1) \mid \mathbf{d}(u) = \text{root}(t_2)\},$$

and identify $\text{Nodes}(t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2))$ with the disjoint union of $\text{Nodes}(t_3)$ and $\text{Nodes}(t_2) \setminus \{\text{root}(t_2)\}$. We denote the map sending the pair (\mathbf{f}, \mathbf{g}) to the pair (\mathbf{d}, \mathbf{e}) as follows:

$$L(\mathbf{f}, \mathbf{g}) = (\mathbf{d}, \mathbf{e}).$$

The following lemma asserts that L and R are inverses of each other.

Lemma 2.6 *Let $t_1, t_2, t_3 \in \text{Tree}(\mathcal{X})$, and let*

$$\begin{aligned} \mathbf{d} & : \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_2) \\ \mathbf{e} & : \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)) \rightarrow \text{Nodes}(t_3) \end{aligned}$$

be functions. Then

$$L \circ R(\mathbf{d}, \mathbf{e}) = (\mathbf{d}, \mathbf{e}).$$

Let

$$\begin{aligned} \mathbf{f} & : \text{DelRoot}(t_2) \rightarrow \text{Nodes}(t_3) \\ \mathbf{g} & : \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2)) \end{aligned}$$

be functions. Then

$$R \circ L(\mathbf{f}, \mathbf{g}) = (\mathbf{f}, \mathbf{g}).$$

PROOF: We first show that $L \circ R(\mathbf{d}, \mathbf{e}) = (\mathbf{d}, \mathbf{e})$. Let $R(\mathbf{d}, \mathbf{e}) = (\mathbf{f}, \mathbf{g})$, and let $L(\mathbf{f}, \mathbf{g}) = (\mathbf{d}', \mathbf{e}')$. To show that $\mathbf{d}' = \mathbf{d}$ we consider two cases: if $\mathbf{d}(s) \neq \text{root}(t_2)$, then $\mathbf{g}(s) = \mathbf{d}(s)$, and $\mathbf{d}'(s) = \mathbf{g}(s)$; if $\mathbf{d}(s) = \text{root}(t_2)$, then $\mathbf{g}(s) = \mathbf{e}(s) \in \text{Nodes}(t_3)$, and $\mathbf{d}'(s) = \text{root}(t_2)$. Therefore, in either case, $\mathbf{d}(s) = \mathbf{d}'(s)$.

We next show that $\mathbf{e}' = \mathbf{e}$. Suppose $s \in \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1))$. There are two cases to consider. In the first case, $s = \iota(s')$ for $s' \in \text{DelRoot}(t_2)$. Then $\mathbf{e}'(s) = \mathbf{f}(s') = \mathbf{e} \circ \iota(s') = \mathbf{e}(s)$. In the second case, $s \in \text{DelRoot}(t_1)$, and $\mathbf{d}(s) = \text{root}(t_2)$. Then $\mathbf{g}(s) = \mathbf{e}(s) \in \text{Nodes}(t_3)$, so $\mathbf{e}'(s) = \mathbf{g}(s) = \mathbf{e}(s)$. Therefore, in either case, $\mathbf{e}'(s) = \mathbf{e}(s)$.

We finally show that $R \circ L(\mathbf{f}, \mathbf{g}) = (\mathbf{f}, \mathbf{g})$. Let $L(\mathbf{f}, \mathbf{g}) = (\mathbf{d}, \mathbf{e})$, and let $R(\mathbf{d}, \mathbf{e}) = (\mathbf{f}', \mathbf{g}')$. If $s \in \text{DelRoot}(t_2)$, then $\mathbf{f}'(s) = \mathbf{e} \circ \iota(s) = \mathbf{f}(s)$. We show that $\mathbf{g}' = \mathbf{g}$ by considering two cases. In the first case, $\mathbf{g}(s) \in \text{Nodes}(t_2) \setminus \{\text{root}(t_2)\}$. Then $\mathbf{d}(s) = \mathbf{g}(s)$, and $\mathbf{d}(s) \neq \text{root}(t_2)$. Therefore $\mathbf{g}'(s) = \mathbf{d}(s) = \mathbf{g}(s)$. In the second case, $\mathbf{g}(s) \in \text{Nodes}(t_3)$. Then $\mathbf{d}(s) = \text{root}(t_2)$. Now, since $s \in \text{DelRoot}(t_1)$ and $\mathbf{e}(s) \in \text{Nodes}(t_3)$, we have that $\mathbf{g}'(s) = \mathbf{e}(s) = \mathbf{g}(s)$. Therefore $\mathbf{g}' = \mathbf{g}$. This completes the proof of the lemma.

Lemma 2.7 *Let $t_1, t_2, t_3 \in \text{Tree}(\mathcal{X})$, let*

$$\begin{aligned} \mathbf{d} &: \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_2) \\ \mathbf{e} &: \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)) \rightarrow \text{Nodes}(t_3) \end{aligned}$$

be functions, and let

$$(\mathbf{f}, \mathbf{g}) = R(\mathbf{d}, \mathbf{e}).$$

Then the underlying trees of

$$t_3 \leftarrow_{\mathbf{e}} \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1))$$

and

$$(t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2)) \leftarrow_{\mathbf{g}} \text{DelRoot}(t_1)$$

are equal.

PROOF: Let

$$u = t_3 \leftarrow_{\mathbf{e}} \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1))$$

and

$$v = (t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2)) \leftarrow_{\mathbf{g}} \text{DelRoot}(t_1).$$

The sets of nodes of the underlying trees of u and v are the same: the disjoint union of $\text{Nodes}(t_3)$, $\text{Nodes}(t_2) \setminus \{\text{root}(t_2)\}$, and $\text{Nodes}(t_1) \setminus \{\text{root}(t_1)\}$. Let α be one of these nodes. We consider the three possible cases. If $\alpha \in \text{Nodes}(t_3)$, then the node α has the same parent in both u and v as it has in t_3 . If $\alpha \in \text{Nodes}(t_2) \setminus \{\text{root}(t_2)\}$, then there are two subcases to consider. If α is not a child of the root of t_2 , then the node α has the same parent in both u and v as it has in t_2 . If α is a child of the root of t_2 , then α is the root of some tree $s \in \text{DelRoot}(t_2)$. In u , the parent of α is $\mathbf{e} \circ \iota(s)$. In v , the parent of α is $\mathbf{f}(s) = \mathbf{e} \circ \iota(s)$. Therefore, in both subcases, α has the same node as parent in both u and v . If $\alpha \in \text{Nodes}(t_1) \setminus \{\text{root}(t_1)\}$, then there are two subcases to consider. If α is not a child of the root of t_1 , then the node α has the same parent in both u and v as it has in t_1 . If α is a child of the root of t_1 , then α is the root of some tree $s \in \text{DelRoot}(t_1)$. In u , if $\mathbf{d}(s) \neq \text{root}(t_2)$, then the parent of α is $\mathbf{d}(s)$; if $\mathbf{d}(s) = \text{root}(t_2)$, then the parent of α is $\mathbf{e}(s)$. In v , the parent of α is $\mathbf{g}(s)$. But $\mathbf{g}(s) = \mathbf{d}(s)$ if $\mathbf{d}(s) \neq \text{root}(t_2)$, and $\mathbf{g}(s) = \mathbf{e}(s)$ if $\mathbf{d}(s) = \text{root}(t_2)$. Therefore, in both subcases, α has the same node as parent in both u and v . So in all cases, the node α has the same parent in the underlying trees of both u and v . It follows that both u and v have the same underlying tree. This completes the proof of the lemma.

The next axiom says that the equality of trees in Lemma 2.7 holds in the family \mathcal{X} .

Axiom 8 *Let \mathcal{X} be a family of trees, let $t_1, t_2, t_3 \in \text{Tree}(\mathcal{X})$, let*

$$\begin{aligned} \mathbf{d} &: \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_2) \\ \mathbf{e} &: \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)) \rightarrow \text{Nodes}(t_3) \end{aligned}$$

be functions, and let

$$(\mathbf{f}, \mathbf{g}) = \mathbf{R}(\mathbf{d}, \mathbf{e}).$$

Then

$$t_3 \leftarrow_{\mathbf{e}} \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)) = (t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2)) \leftarrow_{\mathbf{g}} \text{DelRoot}(t_1).$$

We now develop the material leading up to an axiom necessary for the ‘‘Hopf condition’’ on the Hopf algebra we will associate with the family \mathcal{X} .

Fix $t \in \text{Tree}(\mathcal{X})$ and $X \in \text{Forest}(\mathcal{X})$. Suppose that $\mathbf{d} : X \rightarrow \text{Nodes}(t)$ is a linking function, and that $U \subset \text{DelRoot}(t \leftarrow_{\mathbf{d}} X)$ is a subforest. We define subforests

$$V \subset \text{DelRoot}(t)$$

and

$$W \subset X$$

as follows. Let

$$V = \{ v \in \text{DelRoot}(t) \mid \iota(v) \in U \}$$

and

$$W = \left\{ w \in X \mid \begin{array}{l} \mathbf{d}(w) = \text{root}(t) \text{ and } w \in U, \text{ or} \\ \mathbf{d}(w) \in \text{Nodes}(v) \text{ for some } v \in V \end{array} \right\}$$

and define linking functions

$$\mathbf{e} : W \rightarrow \text{Nodes}(e \leftarrow_1 \text{DelRoot}(t) | V)$$

and

$$\mathbf{f} : X \setminus W \rightarrow \text{Nodes}(e \leftarrow_1 \text{DelRoot}(t) | (\text{DelRoot}(t) \setminus V))$$

as follows:

$$\mathbf{e}(w) = \begin{cases} \text{root}(e) & \text{if } \mathbf{d}(w) = \text{root}(t) \\ \mathbf{d}(w) & \text{if } \mathbf{d}(w) \in \text{Nodes}(v) \text{ for some } v \in V \end{cases}$$

and

$$\mathbf{f}(w) = \begin{cases} \text{root}(e) & \text{if } \mathbf{d}(w) = \text{root}(t) \\ \mathbf{d}(w) & \text{if } \mathbf{d}(w) \in \text{Nodes}(v) \text{ for some } v \in \text{DelRoot}(t) \setminus V. \end{cases}$$

Note that the set over which U ranges depends on \mathbf{d} , and the sets over which \mathbf{e} and \mathbf{f} range depend on V and W .

We denote the map sending (\mathbf{d}, U) to $(V, W, \mathbf{e}, \mathbf{f})$ as follows:

$$M(\mathbf{d}, U) = (V, W, \mathbf{e}, \mathbf{f}).$$

Suppose now that $V \subset \text{DelRoot}(t)$ and $W \subset X$ are subforests, and that

$$\begin{aligned} \mathbf{e} & : W \rightarrow \text{Nodes}(e \leftarrow_1 \text{DelRoot}(t) | V) \\ \mathbf{f} & : X \setminus W \rightarrow \text{Nodes}(e \leftarrow_1 \text{DelRoot}(t) | (\text{DelRoot}(t) \setminus V)) \end{aligned}$$

are linking functions. Let $\mathbf{d} : X \rightarrow \text{Nodes}(t)$ be the linking function defined as follows:

$$\mathbf{d}(x) = \begin{cases} \mathbf{e}(x) & \text{if } x \in W \\ \mathbf{f}(x) & \text{if } x \notin W, \end{cases}$$

and let

$$U = \iota(V) \cup \{x \in W \mid \mathbf{d}(x) = \text{root}(t)\}.$$

(Recall that ι was defined in Equation (2.5).) We denote the map associating (\mathbf{d}, U) to $(V, W, \mathbf{e}, \mathbf{f})$ as follows:

$$D(V, W, \mathbf{e}, \mathbf{f}) = (\mathbf{d}, U).$$

The following lemma asserts that M and D are inverses of each other.

Lemma 2.8 *Suppose $t \in \text{Tree}(\mathcal{X})$ and $X \in \text{Forest}(\mathcal{X})$. Let*

$$\mathbf{d} : X \rightarrow \text{Nodes}(X)$$

be a function, and let $U \subset \text{DelRoot}(t \leftarrow_{\mathbf{d}} X)$ be a subforest. Then

$$D \circ M(\mathbf{d}, U) = (\mathbf{d}, U).$$

Let $V \subset \text{DelRoot}(t)$ and $W \subset X$ be subforests, and

$$\mathbf{e} : W \rightarrow \text{Nodes}(e \leftarrow_1 \text{DelRoot}(t) \mid V)$$

$$\mathbf{f} : X \setminus W \rightarrow \text{Nodes}(e \leftarrow_1 \text{DelRoot}(t) \mid (\text{DelRoot}(t) \setminus V))$$

be functions. Then

$$M \circ D(V, W, \mathbf{e}, \mathbf{f}) = (V, W, \mathbf{e}, \mathbf{f}).$$

PROOF: We first show that $D \circ M(\mathbf{d}, U) = (\mathbf{d}, U)$. Let $M(\mathbf{d}, U) = (V, W, \mathbf{e}, \mathbf{f})$, and let $D(V, W, \mathbf{e}, \mathbf{f}) = (\mathbf{d}', U')$. We first show that $\mathbf{d}' = \mathbf{d}$. Using the identification $t = e \leftarrow_1 \text{DelRoot}(t)$ from Axiom 7, we have that $\text{root}(t) = \text{root}(e)$ in $\text{Nodes}(t) = \text{Nodes}(e \leftarrow_1 \text{DelRoot}(t))$. For $x \in X$, we have that $\mathbf{d}'(x)$ is $\mathbf{e}(x)$ or $\mathbf{f}(x)$, which is $\text{root}(e)$ if $\mathbf{d}(x) = \text{root}(t)$, and $\mathbf{d}(x)$ otherwise. Therefore $\mathbf{d}'(x) = \mathbf{d}(x)$. Since

$$\text{DelRoot}(t \leftarrow_{\mathbf{d}} X) = \iota(\text{DelRoot}(t)) \cup \{x \in X \mid \mathbf{d}(x) = \text{root}(t)\}$$

as forests, we have

$$\begin{aligned} U' &= \iota(V) \cup \{x \in W \mid \mathbf{d}(x) = \text{root}(t)\} \\ &= (\iota(\text{DelRoot}(t)) \cap U) \cup (\{x \in X \mid \mathbf{d}(x) = \text{root}(t)\} \cap U) \\ &= (\iota(\text{DelRoot}(t)) \cup \{x \in X \mid \mathbf{d}(x) = \text{root}(t)\}) \cap U \\ &= \text{DelRoot}(t \leftarrow_{\mathbf{d}} X) \cap U \\ &= U. \end{aligned}$$

Therefore $D \circ M(\mathbf{d}, U) = (\mathbf{d}, U)$.

We now show that

$$M \circ D(V, W, \mathbf{e}, \mathbf{f}) = (V, W, \mathbf{e}, \mathbf{f}).$$

Let $D(V, W, \mathbf{e}, \mathbf{f}) = (\mathbf{d}, U)$, and let $M(\mathbf{d}, U) = (V', W', \mathbf{e}', \mathbf{f}')$. Since

$$\iota(V') = \iota(\text{DelRoot}(t)) \cap U = \iota(V)$$

and ι is injective, it follows that $V' = V$. The definition of \mathbf{d} implies that exactly one of the following possibilities occurs for $x \in X$.

$$\begin{aligned} \mathbf{d}(x) &= \text{root}(e); \\ \mathbf{d}(x) &= \mathbf{e}(x) \in \text{Nodes}(e \leftarrow_1 \text{DelRoot}(t)|V) \setminus \{\text{root}(e)\} \text{ and } x \in W; \\ \mathbf{d}(x) &= \mathbf{f}(x) \in \text{Nodes}(e \leftarrow_1 \text{DelRoot}(t)|(\text{DelRoot}(t) \setminus V)) \setminus \{\text{root}(e)\} \text{ and } x \notin W. \end{aligned}$$

Comparing this with the definition of W' we see that

$$\begin{aligned} W' &= \{x \in W \mid \mathbf{e}(x) = \text{root}(e)\} \cup \\ &\quad \{x \in W \mid \mathbf{e}(x) \in \text{Nodes}(e \leftarrow_1 \text{DelRoot}(t)|V) \setminus \{\text{root}(e)\}\} \\ &= W. \end{aligned}$$

It follows immediately from the definitions of \mathbf{d} , \mathbf{e} , and \mathbf{f} that $\mathbf{e}' = \mathbf{e}$ and $\mathbf{f}' = \mathbf{f}$. This completes the proof of the lemma.

Lemma 2.9 *Let $t \in \text{Tree}(\mathcal{X})$, $X \in \text{Forest}(\mathcal{X})$, $\mathbf{d} : X \rightarrow \text{Nodes}(t)$ be a function, $U \subset \text{DelRoot}(t \leftarrow_{\mathbf{d}} X)$ be a subforest, and*

$$(V, W, \mathbf{e}, \mathbf{f}) = M(\mathbf{d}, U).$$

Then the underlying trees of

$$e \leftarrow_1 \text{DelRoot}(t \leftarrow_{\mathbf{d}} X)|U$$

and

$$(e \leftarrow_1 \text{DelRoot}(t)|V) \leftarrow_{\mathbf{e}} X|W$$

are equal.

PROOF: For simplicity we write $e \leftarrow_1 U$ for $e \leftarrow_1 \text{DelRoot}(t \leftarrow_{\mathbf{d}} X)|U$, etc. We first observe that

$$\begin{aligned}
\text{Nodes}(e \leftarrow_1 U) &= \{\text{root}(e)\} \cup \bigcup_{u \in U} \text{Nodes}(u) \\
&= \{\text{root}(e)\} \cup \bigcup_{u \in U \cap \text{DelRoot}(t)} \text{Nodes}(u) \cup \\
&\quad \bigcup_{u \in U \cap X} \text{Nodes}(u) \\
&= \{\text{root}(e)\} \cup \bigcup_{v \in V} \left(\text{Nodes}(v) \cup \bigcup_{\substack{w \in X \\ \mathbf{d}(w) \in \text{Nodes}(v)}} \text{Nodes}(w) \right) \cup \\
&\quad \bigcup_{u \in U \cap X} \text{Nodes}(u) \\
&= \{\text{root}(e)\} \cup \bigcup_{v \in V} \text{Nodes}(v) \cup \bigcup_{w \in W} \text{Nodes}(w) \\
&= \text{Nodes}((e \leftarrow_1 V) \leftarrow_{\mathbf{e}} W),
\end{aligned}$$

so that the sets of nodes of the underlying trees are the same.

To complete the proof, we show that each node in these two trees has the same parent in both trees. If α is a node of t , then its parent in t is the same as its parent in $e \leftarrow_1 U$ and in $(e \leftarrow_1 V) \leftarrow_{\mathbf{e}} W$. If α is a node of a tree $x \in X$, there are two possibilities. In case α is not the root of x , then its parent in x is the same as its parent in $e \leftarrow_1 U$ and in $(e \leftarrow_1 V) \leftarrow_{\mathbf{e}} W$. In case α is the root of x , its parent in $e \leftarrow_1 U$ is $\mathbf{d}(x)$; its parent in $(e \leftarrow_1 V) \leftarrow_{\mathbf{e}} W$ is $\mathbf{e}(x)$. Making the usual identification of $\text{root}(t)$ with $\text{root}(e)$, we have that $\mathbf{e}(x) = \mathbf{d}(x)$, so that α has the same parent in both trees in this case also. This completes the proof of the lemma.

The next axiom says that the equality of trees in Lemma 2.9 holds in the family \mathcal{X} .

Axiom 9 *Let \mathcal{X} be a family of trees, let $t \in \text{Tree}(\mathcal{X})$, $X \in \text{Forest}(\mathcal{X})$, let $\mathbf{d} : X \rightarrow \text{Nodes}(t)$ be a function, let $U \subset \text{DelRoot}(t \leftarrow_{\mathbf{d}} X)$ be a subforest, and let*

$$(V, W, \mathbf{e}, \mathbf{f}) = \text{M}(\mathbf{d}, U).$$

Then

$$e \leftarrow_1 \text{DelRoot}(t \leftarrow_{\mathbf{d}} X)|U = (e \leftarrow_1 \text{DelRoot}(t)|V) \leftarrow_{\mathbf{e}} X|W.$$

3 The Hopf algebra associated with a family

In this section we describe the graded connected cocommutative Hopf algebra associated with the family \mathcal{X} , and review some facts about the structure of such Hopf algebras.

Suppose that \mathcal{X} is a family satisfying Axioms 1–9 of the previous section. Let k be a field of characteristic 0. Define $k\{\mathcal{X}\}$ to be the vector space over k with basis $\text{Tree}(\mathcal{X})$. We grade $k\{\mathcal{X}\}$ as follows: if the underlying tree of $t \in \text{Tree}(\mathcal{X})$ has $n + 1$ nodes, then t has degree n . By Axiom 6 there is only one $t \in \text{Tree}(\mathcal{X})$ whose underlying tree has one node. Therefore the graded vector space $k\{\mathcal{X}\}$ is connected.

We define a product on $k\{\mathcal{X}\}$ as follows: if $t_1, t_2 \in \text{Tree}(\mathcal{X})$, define

$$t_1 \cdot t_2 = \sum t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1), \quad (3.1)$$

where the sum ranges over all possible linking maps $\mathbf{d} : \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_2)$. We extend this product to all of $k\{\mathcal{X}\}$ by linearity. It is immediate that this product respects the grading we have defined. We now verify that this product is associative. Note that

$$(t_1 \cdot t_2) \cdot t_3 = \sum t_3 \leftarrow_{\mathbf{e}} \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)),$$

where the sum is taken over all pairs (\mathbf{d}, \mathbf{e}) with $\mathbf{d} : \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_2)$, and $\mathbf{e} : \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)) \rightarrow \text{Nodes}(t_3)$, and that

$$t_1 \cdot (t_2 \cdot t_3) = \sum (t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2)) \leftarrow_{\mathbf{g}} \text{DelRoot}(t_1),$$

where the sum is taken over all pairs (\mathbf{f}, \mathbf{g}) with $\mathbf{f} : \text{DelRoot}(t_2) \rightarrow \text{Nodes}(t_3)$, and $\mathbf{g} : \text{DelRoot}(t_2) \rightarrow \text{Nodes}(t_3 \leftarrow_{\mathbf{f}} \text{DelRoot}(t_2))$. Now Lemma 2.6 gives a one-one correspondence between the set of pairs $\{(\mathbf{d}, \mathbf{e})\}$ over which the summation equalling $(t_1 \cdot t_2) \cdot t_3$ is taken, and the set of pairs $\{(\mathbf{f}, \mathbf{g})\}$ over which the summation equalling $t_1 \cdot (t_2 \cdot t_3)$ is taken. Axiom 8 implies that the corresponding terms of the summations are equal. Axiom 7 implies that if $t \in \text{Tree}(\mathcal{X})$, and $e \in \text{Tree}(\mathcal{X})$ is the unique element whose underlying tree has only one node, then $t \cdot e = t$ and $e \cdot t = t$.

The definition of the product given in Equation (3.1) may appear to be reversed. The reason for this apparent reversal is that this product of trees is one which has been used in applications involving data structures representing differential operators (see [5], [6], [7], and [8]). The reversal is similar

to the reversal which occurs in matrix multiplication, in the correspondence between linear transformations on a finite-dimensional vector space and matrices.

We now define a coproduct $\Delta : k\{\mathcal{X}\} \rightarrow k\{\mathcal{X}\} \otimes k\{\mathcal{X}\}$ as follows: if $t \in \text{Tree}(\mathcal{X})$ define

$$\Delta(t) = \sum (e \leftarrow_1 X|U) \otimes (e \leftarrow_1 X|(X \setminus U)),$$

where $X = \text{DelRoot}(t)$, and the sum is taken over all subforests $U \subseteq X$. ($X \setminus Y$ denotes the set-theoretic difference of X and Y .) We extend Δ to all of $k\{\mathcal{X}\}$ by linearity. It is immediate that this coproduct respects the grading on $k\{\mathcal{X}\}$. We now verify that Δ is coassociative. For trees, coassociativity is immediate: all partitions of $\text{DelRoot}(t)$ as a union of three disjoint (possibly empty) sets is achieved either by partitioning it into two disjoint sets, and then partitioning the first set into two disjoint sets, or by partitioning it into two disjoint sets, and then partitioning the second set into two disjoint sets. Axiom 4 implies that this partitioning in two different ways is equivalent in $\text{Forest}(\mathcal{X})$, and Axiom 7 implies that we have

$$(\Delta \otimes I) \circ \Delta = (I \otimes \Delta) \circ \Delta$$

as maps from $k\{\mathcal{X}\}$ to $k\{\mathcal{X}\} \otimes k\{\mathcal{X}\} \otimes k\{\mathcal{X}\}$. The counit $\epsilon : k\{\mathcal{X}\} \rightarrow k$ is defined as follows:

$$\epsilon(t) = \begin{cases} 1 & \text{if } t = e \\ 0 & \text{otherwise.} \end{cases}$$

We extend ϵ to all of $k\{\mathcal{X}\}$ by linearity. It follows that

$$(\epsilon \otimes I) \circ \Delta = (I \otimes \epsilon) \circ \Delta = I$$

from Axioms 2 and 7. It follows from the fact that the set of all subsets of a set equals the set of all complements of subsets of a set that this coalgebra is cocommutative.

We now prove that the map $\Delta : k\{\mathcal{X}\} \rightarrow k\{\mathcal{X}\} \otimes k\{\mathcal{X}\}$ is an algebra homomorphism, that is, that

$$\Delta(t_1 \cdot t_2) = \Delta(t_1) \cdot \Delta(t_2)$$

for $t_1, t_2 \in \text{Tree}(\mathcal{X})$. We compute

$$\begin{aligned} \Delta(t_1 \cdot t_2) &= \sum_{\mathbf{d}: X \rightarrow \text{Nodes}(t_2)} \Delta(t_2 \leftarrow_{\mathbf{d}} X) \\ &= \sum (e \leftarrow_1 Z_{\mathbf{d}}|U) \otimes (e \leftarrow_1 Z_{\mathbf{d}}|(Z_{\mathbf{d}} \setminus U)), \end{aligned}$$

where $Z_{\mathbf{d}} = \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1))$, and the second sum ranges over all pairs (\mathbf{d}, U) , with $\mathbf{d} : X \rightarrow \text{Nodes}(t_2)$ and $U \subset \text{DelRoot}(t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1))$. On the other hand

$$\begin{aligned} \Delta(t_1) \cdot \Delta(t_2) &= \left(\sum_{V \subset X} (e \leftarrow_{\mathbf{1}} X|V) \otimes (e \leftarrow_{\mathbf{1}} X|(X \setminus V)) \right) \cdot \\ &\quad \left(\sum_{W \subset Y} (e \leftarrow_{\mathbf{1}} Y|W) \otimes (e \leftarrow_{\mathbf{1}} Y|(Y \setminus W)) \right) \\ &= \sum (e \leftarrow_{\mathbf{1}} Y|W) \leftarrow_{\mathbf{e}} X|V \otimes (e \leftarrow_{\mathbf{1}} Y|(Y \setminus W)) \leftarrow_{\mathbf{f}} X|(X \setminus V), \end{aligned}$$

where $X = \text{DelRoot}(t_1)$, $Y = \text{DelRoot}(t_2)$, and the last sum is taken over all quadruples $(V, W, \mathbf{e}, \mathbf{f})$ with $V \subseteq X$, $W \subseteq Y$, $\mathbf{e} : X|V \rightarrow \text{Nodes}(e \leftarrow_{\mathbf{1}} Y|W)$, and $\mathbf{f} : X|(X \setminus V) \rightarrow \text{Nodes}(e \leftarrow_{\mathbf{1}} Y|(Y \setminus W))$. Now Lemma 2.8 gives a one-one correspondence between the terms of the summations equalling $\Delta(t_1 \cdot t_2)$ and $\Delta(t_1) \cdot \Delta(t_2)$. Axiom 9 implies that the corresponding terms in the summations are equal.

We summarize this discussion in the following theorem.

Theorem 3.2 *Let \mathcal{X} be a family of trees satisfying Axioms 1–9. Then $k\{\mathcal{X}\}$ is a cocommutative graded connected Hopf algebra.*

If A is a Hopf algebra, then the *primitive* elements of A are defined

$$P(A) = \{ a \in A \mid \Delta(a) = 1 \otimes a + a \otimes 1 \}.$$

It can be shown that $P(A)$ is a Lie subalgebra of A^- , which is the Lie algebra with the same underlying vector space as the associative algebra A , and in which the bracket operation is defined by $[a, b] = ab - ba$.

If L is a Lie algebra, then the universal enveloping algebra $U(L)$ is a Hopf algebra. If $x \in L$, then $\Delta(x) = 1 \otimes x + x \otimes 1$, and $\epsilon(x) = 0$. The maps Δ and ϵ are extended to all of $U(L)$ using the facts that Δ is an algebra homomorphism, and that L generates $U(L)$ as an algebra. The following theorem gives a basis for $U(L)$ in terms of an ordered basis for L .

Theorem 3.3 (Poincaré-Birkhoff-Witt) *Let L be a Lie algebra with ordered basis x_1, \dots, x_n, \dots . Then*

$$\{ x_{i_1}^{e_1} \cdots x_{i_t}^{e_t} \mid i_1 < \cdots < i_t; \ e_k > 0 \}$$

is a basis for $U(L)$.

See [9, page 159] for a proof.

Theorem 3.4 (Milnor-Moore) *Let A be a cocommutative graded connected Hopf algebra. Then*

$$A \cong U(P(A))$$

as Hopf algebras.

See [12, page 244] or [15, page 274] for a proof.

If X is a set, denote by $k\langle X \rangle$ the free associative algebra over k generated by X . Then $k\langle X \rangle$ is a cocommutative Hopf algebra, with $\Delta(x) = 1 \otimes x + x \otimes 1$ for $x \in X$. It can be shown that $P(k\langle X \rangle)$ is the free Lie algebra generated by X .

4 The family of labeled trees

In this section we discuss the structure of $k\{\mathcal{LT}(E_1, \dots, E_M)\}$. Note that if $M = 1$ we are essentially discussing $k\{\mathcal{T}\}$. We give a description of $P(k\{\mathcal{LT}(E_1, \dots, E_M)\})$ and use it, together with the Milnor-Moore Theorem and the Poincaré-Birkhoff-Witt Theorem, to give a new proof of the recurrence relation for the number of rooted trees with n nodes first given by Cayley [3] in 1857.

Let $\mathcal{LT}_1(E_1, \dots, E_M)$ be the set of labeled trees $t \in \mathcal{LT}(E_1, \dots, E_M)$ whose root has only one child.

Theorem 4.1 *The set $\mathcal{LT}_1(E_1, \dots, E_M)$ is a basis for $P(k\{\mathcal{LT}(E_1, \dots, E_M)\})$.*

PROOF: Denote $\mathcal{LT}(E_1, \dots, E_M)$ by \mathcal{LT} , and $\mathcal{LT}_1(E_1, \dots, E_M)$ by \mathcal{LT}_1 . It is easily checked that if $t \in \mathcal{LT}_1$, then $t \in P(k\{\mathcal{LT}\})$. We now show that the elements of \mathcal{LT}_1 span $P(k\{\mathcal{LT}\})$. Define

$$\pi : k\{\mathcal{LT}\} \otimes k\{\mathcal{LT}\} \rightarrow k\{\mathcal{LT}\}$$

as follows: if $t_1, t_2 \in \mathcal{LT}$, let $\pi(t_1 \otimes t_2)$ be the element of \mathcal{LT} formed by identifying the roots of t_1 and t_2 . It is easily checked that if the root of $t \in \mathcal{LT}$ has r children, then $\pi \circ \Delta(t) = 2^r t$. On the other hand, if $a = \sum a_t t \in P(k\{\mathcal{LT}\})$, then $\pi \circ \Delta(a) = 2a$. Since the elements of \mathcal{LT} are linearly independent, it follows that $a_t = 0$ if the root of t has more than one child. This completes the proof of the theorem.

Theorem 4.2 *Let t_n be the number of rooted trees with n nodes, with all nodes but the root labeled using the formal symbols $\{E_1, \dots, E_M\}$. Then*

$$\begin{aligned} t_1 &= 1 \\ t_{n+1} &= \sum_{m_1+2m_2+\dots+rm_r=n} \binom{m_1+Mt_1-1}{m_1} \dots \binom{m_r+Mt_r-1}{m_r}. \end{aligned}$$

PROOF: Let

$$a_n = \dim k\{\mathcal{LT}(E_1, \dots, E_M)\}_n$$

and let

$$p_n = \dim P(k\{\mathcal{LT}(E_1, \dots, E_M)\})_n.$$

The definition of the grading on $k\{\mathcal{X}\}$ implies that $t_{n+1} = a_n$. Theorem 4.1 implies that $p_n = Ma_{n-1}$. Since the number of monomials of length m of the form $x_1^{e_1} \dots x_{p_t}^{e_{p_t}}$, where $\{x_i\}$ is an ordered basis of $P(k\{\mathcal{LT}(E_1, \dots, E_M)\})_t$, is $\binom{m+p_t-1}{m}$, it follows that

$$\begin{aligned} a_n &= \sum_{m_1+2m_2+\dots+rm_r=n} \binom{m_1+p_1-1}{m_1} \dots \binom{m_r+p_r-1}{m_r} \\ &= \sum_{m_1+2m_2+\dots+rm_r=n} \binom{m_1+Ma_0-1}{m_1} \dots \binom{m_r+Ma_{r-1}-1}{m_r}. \end{aligned}$$

The statement of the theorem follows immediately from this.

The following result (with $M = 1$) was proved by Cayley [3] in 1857.

Corollary 4.3 *Let $T(z) = \sum_{n=1}^{\infty} t_n z^n$. Then*

$$T(z) = z \prod_{n=1}^{\infty} (1 - z^n)^{-Mt_n}.$$

PROOF: This follows immediately from the Theorem 4.2 upon observing that

$$\binom{r+k-1}{k} = (-1)^k \binom{-r}{k}.$$

5 The family of labeled ordered trees

In this section we discuss the structure of $k\{\mathcal{LOT}(E_1, \dots, E_M)\}$. We will show that this Hopf algebra is isomorphic to the free associative algebra generated by $\mathcal{LOT}_1(E_1, \dots, E_M)$, the set of labeled ordered rooted trees whose root has exactly one child. This fact allows us to give a recurrence relation for the number of labeled ordered rooted trees. This recurrence can be solved to get the number of ordered rooted trees with n nodes. This number can be shown [11] to be the same as the number of binary trees with $n - 1$ nodes, which was given by Catalan [2] in 1838, and by Cayley [4] in 1859.

Theorem 5.1

$$k\{\mathcal{LOT}(E_1, \dots, E_M)\} \cong k\langle \mathcal{LOT}_1(E_1, \dots, E_M) \rangle.$$

PROOF: Write \mathcal{LOT} for $\mathcal{LOT}(E_1, \dots, E_M)$, and \mathcal{LOT}_1 for $\mathcal{LOT}_1(E_1, \dots, E_M)$. We introduce a filtration on $k\langle \mathcal{LOT}_1 \rangle$ by defining $F_p k\langle \mathcal{LOT}_1 \rangle$ to be the subspace of $k\langle \mathcal{LOT}_1 \rangle$ spanned by all monomials of length $\leq p$. It is clear that

$$\begin{aligned} F_{-1} k\langle \mathcal{LOT}_1 \rangle &= 0, \\ \bigcup_p F_p k\langle \mathcal{LOT}_1 \rangle &= k\langle \mathcal{LOT}_1 \rangle, \\ (F_p k\langle \mathcal{LOT}_1 \rangle) \cdot (F_q k\langle \mathcal{LOT}_1 \rangle) &\subseteq F_{p+q} k\langle \mathcal{LOT}_1 \rangle. \end{aligned}$$

We introduce a filtration on $k\{\mathcal{LOT}\}$ by defining $F_p k\{\mathcal{LOT}\}$ to be the subspace spanned by all trees whose root has p or fewer children. This filtration satisfies

$$\begin{aligned} F_{-1} k\{\mathcal{LOT}\} &= 0, \\ \bigcup_p F_p k\{\mathcal{LOT}\} &= k\{\mathcal{LOT}\}, \\ (F_p k\{\mathcal{LOT}\}) \cdot (F_q k\{\mathcal{LOT}\}) &\subseteq F_{p+q} k\{\mathcal{LOT}\}. \end{aligned}$$

More precisely, if the root of t_1 has p children, and the root of t_2 has q children, then

$$t_1 \cdot t_2 = \sum_{\mathbf{d}: \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_2)} t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1),$$

and the root of $t_2 \leftarrow_{\mathbf{d}} \text{DelRoot}(t_1)$ has $q + r$ children, where r is the number of $x \in \text{DelRoot}(t_1)$ satisfying $\mathbf{d}(x) = \text{root}(t_2)$. Let $\mathbf{d}_0 : \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_2)$ be defined by $\mathbf{d}_0(x) = \text{root}(t_2)$ for all $x \in \text{DelRoot}(t_1)$. Then

$$t_1 \cdot t_2 = t_2 \leftarrow_{\mathbf{d}_0} \text{DelRoot}(t_1) + \tau,$$

where $\tau \in F_{p+q-1}k\{\mathcal{LOT}\}$.

There is a unique algebra homomorphism $\phi : k\langle \mathcal{LOT}_1 \rangle \rightarrow k\{\mathcal{LOT}\}$ which is the identity on $\mathcal{LOT}_1 \subset \mathcal{LOT}$. It is clear that $\phi(F_p k\langle \mathcal{LOT}_1 \rangle) \subseteq F_p k\{\mathcal{LOT}\}$. More precisely, if $x_1, \dots, x_p \in \mathcal{LOT}_1$, then $\phi(x_1 \cdots x_p) = \phi(x_1) \cdots \phi(x_p)$ is congruent modulo $F_{p-1}k\{\mathcal{LOT}\}$ to the tree whose root has exactly the following p children: the children of the roots of x_p, \dots, x_1 in that order. Therefore ϕ induces an isomorphism

$$F_p k\langle \mathcal{LOT}_1 \rangle / F_{p-1} k\langle \mathcal{LOT}_1 \rangle \rightarrow F_p k\{\mathcal{LOT}\} / F_{p-1} k\{\mathcal{LOT}\}.$$

It follows that ϕ is an isomorphism. This completes the proof of the theorem.

Theorem 5.2 *Let t_n be the number of ordered rooted trees with n nodes, with all of the nodes but the root labeled using the formal symbols $\{E_1, \dots, E_M\}$. Then*

$$\begin{aligned} t_1 &= 1 \\ t_{n+1} &= M(t_1 t_n + t_2 t_{n-1} + \cdots + t_n t_1). \end{aligned}$$

PROOF: Let

$$a_n = \dim k\{\mathcal{LOT}(E_1, \dots, E_M)\}_n,$$

and

$$g_n = \text{Card}\{t \in \mathcal{LOT}_1(E_1, \dots, E_M) \mid t \text{ has } n+1 \text{ nodes}\},$$

the number of free generators of degree n . The definition of the grading on $k\{\mathcal{X}\}$ implies that $t_{n+1} = a_n$. It follows from the fact that $k\{\mathcal{LOT}\} \cong k\langle \mathcal{LOT}_1 \rangle$ that $g_n = M a_{n-1}$, so

$$\begin{aligned} a_n &= g_1 a_{n-1} + \cdots + g_n a_0 \\ &= M(a_0 a_{n-1} + \cdots + a_{n-1} a_0), \end{aligned}$$

or

$$t_{n+1} = M(t_1 t_n + \cdots + t_n t_1).$$

This completes the proof of the theorem.

Corollary 5.3 *Let t_n be the number of ordered rooted trees with n nodes, with all of the nodes but the root labeled using the formal symbols $\{E_1, \dots, E_M\}$. Then*

$$t_n = \frac{M^{n-1}}{n} \binom{2n-2}{n-1}.$$

PROOF: The solution of the recurrence in Theorem 5.2 is adapted from [11, pages 388–389]. This proof first appears in [1]. Let

$$T(z) = \sum_{n=1}^{\infty} t_n z^n.$$

Then Theorem 5.2 implies that

$$T(z) = MT(z)^2 + z$$

which, together with the fact that the coefficient of z in $T(z)$ equals 1 implies that

$$T(z) = \frac{1 - \sqrt{1 - 4Mz}}{2M}.$$

Now

$$\sqrt{1 - 4Mz} = \sum_{n=0}^{\infty} (-1)^n 4^n M^n \binom{\frac{1}{2}}{n} z^n$$

and

$$\binom{\frac{1}{2}}{n} = \frac{(-1)^{n-1}}{2^{2n}(2n-1)} \binom{2n}{n},$$

so

$$T(z) = \sum_{n=1}^{\infty} \frac{M^{n-1}}{n} \binom{2n-2}{n-1} z^n,$$

which proves the corollary.

6 Heap-ordered trees and other families

In this section, we describe some other families of trees related to \mathcal{OT} , and prove a generalization of Theorem 5.1 for these families. We then apply this result to get a recurrence relation on the number of indecomposable heap-ordered trees. First we give a description of the category of families of trees.

If \mathcal{X} and \mathcal{Y} are families of trees, we define a morphism $f : \mathcal{X} \rightarrow \mathcal{Y}$ to be a pair of maps $\text{Tree}(\mathcal{X}) \rightarrow \text{Tree}(\mathcal{Y})$ and $\text{Forest}(\mathcal{X}) \rightarrow \text{Forest}(\mathcal{Y})$ which are the identity on the underlying trees and forests, and which commute with DelRoot , $|$, and \leftarrow . We thus define the category **Family** of families of trees. Note that \mathcal{T} is a terminal object in **Family**. In Section 3 we defined the functor $k\{-\}$ from **Family** to the category of cocommutative graded connected Hopf algebras over k .

In the example below, we will speak of generations of a node in a tree: the singleton $\{\alpha\}$ is the first generation of α ; if $\{\beta_1, \dots, \beta_k\}$ is the n^{th} generation of the node α , then the $n + 1^{\text{st}}$ generation of α is the set of children of the nodes β_1, \dots, β_k .

Example 6.1 *The family \mathcal{GT}_n of n -generation ordered trees. The set $\text{Tree}(\mathcal{GT}_n)$ is the set of finite rooted trees, together with, for each node α , a total ordering on the set X_α consisting of the nodes which constitute the first n generations of α . The ordering on $X_\alpha \cap X_\beta$ must be the same, whether induced as a subset of X_α or of X_β . The orderings must also satisfy the condition that each node α precede all of its descendants in X_α .*

The set $\text{Forest}(\mathcal{GT}_n)$ is the set of finite forests of finite rooted trees, together with, for each node α , a total ordering on the set X_α consisting of the nodes which constitute the first n generations of α , and a total ordering on the set Y consisting of the nodes which constitute the first $n - 1$ generations of the nodes which are the roots of the trees in the forest. The ordering on $X_\alpha \cap X_\beta$ (or on $X_\alpha \cap Y$) must be the same, whether induced as a subset of X_α or of X_β (or of Y). The orderings must also satisfy the condition that each node α precede all of its descendants in X_α .

The map DelRoot sends each tree into the forest formed by deleting its root, with the orderings on the sets of nodes given in the obvious way.

If $V \in \text{Forest}(\mathcal{GT}_n)$ and $U \subseteq V$ is a subforest, then $V|U$ is the forest U , with the same orderings on the sets X_α as in V , and with the ordering on the set Y_U associated with U being the one induced by the fact that it is a subset of the set Y_V associated with V .

If $t \in \text{Tree}(\mathcal{GT}_n)$, $U \in \text{Forest}(\mathcal{GT}_n)$, and $\mathbf{d} : U \rightarrow \text{Nodes}(t)$, then $t \leftarrow_{\mathbf{d}} U$ is the tree formed by linking the root of each tree in U to the node $\mathbf{d}(u)$ of t . If α was originally a node of t , the new set X_α is ordered as follows: the original descendants of α in t , up to the n^{th} generation, preserve their original order in the new X_α ; the descendants of α , up to the n^{th} generation, which were among the nodes of trees in U , preserve the order they originally

had in Y_U ; all of the original nodes of t which are descendants of α , up to the n^{th} generation, precede all of the descendants of α , up to the n^{th} generation, which were originally nodes of trees in U . If α was originally a node of a tree in U , the set X_α is unchanged.

Example 6.2 The family \mathcal{LGT}_n of labeled n -generation ordered trees. This is formed by extending the definition of the family \mathcal{GT}_n in the same way that the definition of \mathcal{T} was extended in Example 2.4 to a definition of \mathcal{LT} .

In some cases, the family \mathcal{GT}_n is isomorphic to a previously defined family. The family \mathcal{GT}_1 is isomorphic to the family \mathcal{T} . The family \mathcal{GT}_2 is isomorphic to the family \mathcal{OT} (the map $\mathcal{OT} \rightarrow \mathcal{GT}_2$ is given by: if $t \in \mathcal{OT}$, construct orderings on the sets of nodes and their children by requiring that each node precede all of its children, and keep the ordering of the children unchanged; the inverse map $\mathcal{GT}_2 \rightarrow \mathcal{OT}$ is given by: if $t \in \mathcal{GT}_2$, order the children of each node in the same way that they are ordered in \mathcal{GT}_2).

There is a morphism $\mathcal{GT}_n \rightarrow \mathcal{GT}_{n-1}$ given by restricting the ordering on the set of the first n generations of each node α to the first $n-1$ generations. These morphisms are surjections. It is clear that $\mathcal{HOT} = \varprojlim \mathcal{GT}_n$.

Let \mathcal{X} be a family of trees. If $t_1, t_2 \in \text{Tree}(\mathcal{X})$, write $t_1 \odot t_2$ for the tree

$$t_2 \leftarrow_{\mathbf{d}_0} \text{DelRoot}(t_1),$$

where $\mathbf{d}_0 : \text{DelRoot}(t_1) \rightarrow \text{Nodes}(t_2)$ is the function defined by $\mathbf{d}_0(w) = \text{root}(t_2)$, for all $w \in \text{DelRoot}(t_1)$. Note that Axiom 8 implies that \odot is associative. If $t \in \text{Tree}(\mathcal{X})$ with $t \neq e$, we say that t is *indecomposable* if $t = t_1 \odot t_2$ implies that $t_1 = e$ or $t_2 = e$. Let

$$I(\mathcal{X}) = \{t \in \text{Tree}(\mathcal{X}) \mid t \text{ is indecomposable}\}.$$

By finiteness, every tree $t \in \text{Tree}(\mathcal{X})$ can be written

$$t = t_1 \odot \cdots \odot t_p,$$

with $t_i \in I(\mathcal{X})$.

For example, if $\mathcal{X} = \mathcal{T}$ or \mathcal{OT} , the indecomposable trees are those trees whose roots have exactly one child. In \mathcal{HOT} , writing $t = t_1 \odot t_2$ corresponds to partitioning $\text{DelRoot}(t)$ into two disjoint subforests $V = \text{DelRoot}(t_1)$ and $W = \text{DelRoot}(t_2)$, such that every node of every tree in V precedes every

node of every tree in W . An indecomposable tree $t \in \text{Tree}(\mathcal{HOT})$ is therefore one for which $\text{DelRoot}(t)$ cannot be so partitioned. There is an analogous (but not so clearly expressible) description of indecomposable trees in \mathcal{GT}_n for $n > 2$.

The family \mathcal{X} is said to have the *unique decomposition property* if, whenever

$$x_1 \odot \cdots \odot x_p = y_1 \odot \cdots \odot y_q,$$

with $x_i, y_j \in I(\mathcal{X})$, the ordered sequences (x_1, \dots, x_p) and (y_1, \dots, y_q) are equal. Essentially, the unique decomposition property is a non-commutative unique factorization condition. Note that the families \mathcal{OT} , \mathcal{HOT} , and \mathcal{GT}_n , for $n \geq 2$, all have the unique decomposition property. The family \mathcal{T} does not have the unique decomposition property, because \odot is commutative in \mathcal{T} .

The following theorem can be thought of as a generalization of Theorem 5.1.

Theorem 6.3 *Let \mathcal{X} be a family of trees with the unique decomposition property. Then*

$$k\{\mathcal{X}\} \cong k\langle I(\mathcal{X}) \rangle.$$

PROOF: We introduce a filtration on $k\langle I(\mathcal{X}) \rangle$ by filtering $I(\mathcal{X})$: if $t \in I(\mathcal{X})$, we say that $t \in F_p I(\mathcal{X})$ if the root of t has $\leq p$ children. Next define $F_p k\langle I(\mathcal{X}) \rangle$ to be the subspace spanned by the monomials $x_1 \cdots x_k$, where $x_i \in F_{p_i} I(\mathcal{X})$ and $p_1 + \cdots + p_k \leq p$. It is clear that

$$\begin{aligned} F_{-1} k\langle I(\mathcal{X}) \rangle &= 0, \\ \bigcup_p F_p k\langle I(\mathcal{X}) \rangle &= k\langle I(\mathcal{X}) \rangle, \\ (F_p k\langle I(\mathcal{X}) \rangle) \cdot (F_q k\langle I(\mathcal{X}) \rangle) &\subseteq F_{p+q} k\langle I(\mathcal{X}) \rangle. \end{aligned}$$

We next filter $k\{\mathcal{X}\}$ by defining $F_p k\{\mathcal{X}\}$ to be the subspace spanned by the $t \in \mathcal{X}$ whose roots have $\leq p$ children. This filtration satisfies

$$\begin{aligned} F_{-1} k\{\mathcal{X}\} &= 0, \\ \bigcup_p F_p k\{\mathcal{X}\} &= k\{\mathcal{X}\}, \\ (F_p k\{\mathcal{X}\}) \cdot (F_q k\{\mathcal{X}\}) &\subseteq F_{p+q} k\{\mathcal{X}\}. \end{aligned}$$

There is a unique algebra homomorphism $\phi : k\langle I(\mathcal{X}) \rangle \rightarrow k\{\mathcal{X}\}$ which is the identity on $I(\mathcal{X}) \subset \mathcal{X}$. It is easily checked that $\phi(F_p k\langle I(\mathcal{X}) \rangle) \subseteq F_p k\{\mathcal{X}\}$. Therefore ϕ induces a map

$$\bar{\phi} : F_p k\langle I(\mathcal{X}) \rangle / F_{p-1} k\langle I(\mathcal{X}) \rangle \rightarrow F_p k\{\mathcal{X}\} / F_{p-1} k\{\mathcal{X}\}.$$

If $x_1, \dots, x_k \in I(\mathcal{X})$, $x_1 \cdots x_k \in F_p k\langle I(\mathcal{X}) \rangle$, and $x_1 \cdots x_k \notin F_{p-1} k\langle I(\mathcal{X}) \rangle$, then $\phi(x_1 \cdots x_k) = \phi(x_1) \cdots \phi(x_k)$ is congruent modulo $F_{p-1} k\{\mathcal{X}\}$ to $x_1 \odot \cdots \odot x_k$. Since \mathcal{X} has the unique decomposition property, the elements $x_1 \odot \cdots \odot x_k$ for which $x_1 \cdots x_k \in F_p k\langle I(\mathcal{X}) \rangle$, and $x_1 \cdots x_k \notin F_{p-1} k\langle I(\mathcal{X}) \rangle$, are linearly independent modulo $F_{p-1} k\{\mathcal{X}\}$. Therefore, the map $\bar{\phi}$ is injective. It follows from the fact that every $x \in \mathcal{X}$ can be written $x = x_1 \odot \cdots \odot x_k$, with $x_i \in I(\mathcal{X})$, that $\bar{\phi}$ is surjective. It follows that ϕ is an isomorphism. This completes the proof of the theorem.

It can be shown that a Hopf algebra structure can also be defined on $k\{\mathcal{X}\}$ using \odot as product, rather than the product defined in Section 3. This Hopf algebra is just the associated graded Hopf algebra which is constructed from the filtration $F_p k\{\mathcal{X}\}$ defined in the proof of Theorem 6.3. We summarize this in the following corollary.

Corollary 6.4 *The vector space $k\{\mathcal{X}\}$, with the coproduct and counit defined in Section 3, and with product \odot , is a cocommutative bigraded connected Hopf algebra which is isomorphic to*

$$\sum_{p \geq 0} F_p k\{\mathcal{X}\} / F_{p-1} k\{\mathcal{X}\}.$$

We now use Theorem 6.3 to give a recurrence relation on the number of indecomposable heap-ordered trees. We first compute the number of heap-ordered trees.

Lemma 6.5 *The number of distinct heap-ordered trees with $n + 1$ nodes is $n!$.*

PROOF: The assertion of the lemma is clear for $n = 0$. If $n > 0$, a heap-ordered tree with $n + 1$ nodes can be formed by linking the $n + 1^{\text{st}}$ node, which is last in the total ordering on the nodes, to any of the n nodes in any of the $(n - 1)!$ heap-ordered trees with n nodes. It is clear that the resulting $n!$ heap-ordered trees are all distinct. This completes the proof of the lemma.

Corollary 6.6 *The number g_n of distinct indecomposable heap-ordered trees with $n + 1$ nodes satisfies*

$$\begin{aligned} g_1 &= 1 \\ g_n &= n! - (n-1)!g_1 - \cdots - 1!g_{n-1}. \end{aligned}$$

PROOF: Let t_n be the number of heap-ordered trees with $n + 1$ nodes. By Theorem 6.3, the algebra $k\{\mathcal{HOT}\}$ is freely generated by $I(\mathcal{HOT})$, so

$$t_n = g_1 t_{n-1} + \cdots + g_n t_0.$$

Since $t_n = n!$ by Lemma 6.5, the corollary follows.

References

- [1] M. J. Binet, *Réflexions sur le problème de déterminer le nombre de manières dont une figure rectiligne peut être partagée en triangles au moyen de ses diagonals*, J. Math. Pures Appl. (1) **4** (1839), 79–90.
- [2] E. Catalan, *Note sur une équation aux différences finies*, J. Math. Pures Appl. (1) **3** (1838), 508–516.
- [3] A. Cayley, *On the theory of the analytical forms called trees*, in “Collected Mathematical Papers of Arthur Cayley,” Cambridge Univ. Press, Cambridge, 1890, **3**, 242–246.
- [4] A. Cayley, *On the analytical forms called trees. Second part*, in “Collected Mathematical Papers of Arthur Cayley,” Cambridge Univ. Press, Cambridge, 1891, **4**, 112–115.
- [5] R. Grossman, *Evaluation of expressions involving higher order derivations*, Center for Pure and Applied Mathematics, PAM–367, University of California, Berkeley.
- [6] R. Grossman and R. G. Larson, *Labeled trees and the algebra of differential operators*, Center for Pure and Applied Mathematics, PAM–368, University of California, Berkeley.

- [7] R. Grossman and R. G. Larson, *The symbolic computation of higher-order derivations: symmetries of expressions and actions of group algebras*, Center for Pure and Applied Mathematics, PAM-370, University of California, Berkeley.
- [8] R. Grossman and R. G. Larson, *Labeled trees and the efficient computation of derivations*, in preparation.
- [9] N. Jacobson, “Lie algebras,” Interscience, New York, 1962.
- [10] S. A. Joni and G.-C. Rota, *Coalgebras and bialgebras in combinatorics*, Stud. Appl. Math. **61** (1979), 93–139; reprinted in “Umbral Calculus and Hopf Algebras,” Amer. Math. Soc., Providence, 1982, 1–47.
- [11] D. E. Knuth, “The Art of Computer Programming, Vol. I, Fundamental Algorithms,” 2nd Ed., Addison-Wesley, Reading, 1973.
- [12] J. W. Milnor and J. C. Moore, *On the structure of Hopf algebras*, Ann. Math. (2) **81** (1965), 211–264.
- [13] W. Nichols and M. Sweedler, *Hopf algebras and combinatorics*, in “Umbral Calculus and Hopf Algebras,” Amer. Math. Soc., Providence, 1982, 49–84.
- [14] B. Pareigis, “Four Lectures on Hopf Algebras,” Centre de Recerca Matemàtica, Institut d’Estudis Catalans, No. 6, Octubre 1984.
- [15] M. Sweedler, “Hopf Algebras,” W. A. Benjamin, New York, 1969.
- [16] R. E. Tarjan, “Data Structures and Network Algorithms,” SIAM, Philadelphia, 1983.